

# **Building Cost Effective High Performance 100 Gbps Firewalls**

Jordan A. Caraballo-Vega<sup>1</sup>, George Rumney<sup>2</sup>, John Jasen<sup>2</sup>

<sup>1</sup>University of Puerto Rico at Humacao, Department of Mathematics <sup>2</sup>High Performance Computing, NASA Goddard Space Flight Center, Mail Code 606.2



# Objectives

The continuous growth of the NASA Center for Climate Simulation (NCCS) requires providing high performance security tools and the enhancement of the network capacity. In order to support the requirements of emerging services, the NCCS security team has proposed an architecture to provide extremely cost effective 100 Gbps Firewalls. The aim of this project is to:

- Create a commodity based platform that can process enough packets per second (pps) to sustain a 100 Gbps workload
- Establish a test domain capable of sending massive amounts of pps in order to saturate the system.
- Determine an operating system capable of processing more pps right out of the box.
- Define tuning variables needed to boost scores and decrease interrupts.



Figure 1. NCCS High Performance **Computational Environment** 

Figure 6. The NCCS currently administrates and develops multiple services such as ADAPT, which is responsible for processing huge amounts of data daily for the use of our scientists. It is therefore our responsibility to secure these systems.

# Test Environment

# Client/ Server Interface



**Systems:** The basic components of the test interface include five clients and five servers, for a total of eight Dell R6100 and two Dell R420. Each one is equipped with a 10G network interface.

Hardware – NIC's: Clients and servers are equipped with 10G Intel 8259x and 10G Mellanox ConnectX2 NIC's. Each one has 10G optic cables.



performance (Figure 8).

[ \$i -ge \$ncpu ] && i=18

Figure 9. (A) Representation of /boot/loader.conf file, while (B) "top -CHIPSu" CPU usage output during test.

number and size of rx and tx queues,

pauses, flowcontrol, and the buffer size

need to be changed. Also, balancing the

load of interrupts in the system to an

specific amount of CPUs improve

Figure 8. Bash script to move interrupts to the CPUs

## Netmap & Netmap-fwd

Netmap is a framework for high speed packet I/O implemented as a single, non intrusive kernel module. Together with the netmap-fwd API can easily reach line rate on 10G NIC's (14.88 Mpps).

# Why Firewalls at the NCCS? The NCCS mission is to provide advanced,

simulations.

useable, agile, and efficient high performance

computing services to a wide community of

climate scientists. With more than 4,000

computing nodes, the NCCS computational

environment has become a powerful tool to

provide advanced computing, storage, and

data services to better utilize models and



Figure 2. Simulation of Hurricane Sandy run on the Discover supercomputer at the NCCS. William Putman NASA/Goddard

In order to provide a secure environment for our users, firewalls have been deployed through all our facilities. This network security system, either hardware or software based, grants or rejects network access to traffic flows between untrusted zones and trusted zones. Its role is to help screen out hackers, viruses, and worms from your working interface.



Figure 4. The NCCS currently possess nearly 90,000 processors cores.

### Previous work done by John Jasen at the NCCS reached about ~4M pps, or ~40Gbps firewalls. The test environment consisted of

Transmission Control Protocol (TCP): transmission

that establishes a connection between two hosts

in order to exchange streams of data. It guarantees

delivery of data in the exact order they were sent.

User Datagram Protocol (UDP): it is an alternative

faster transmission that does not provide error

checking. Described as an unreliable and

connectionless protocol, it is mainly used for video

conferencing and real-time computer games.

**Figure 3. NCCS HPC Platform** 

a R820 system built with FreeBSD 10-STABLE and two Chelsio T-580-CR 40Gb ethernet cards. One of the new implementations of our current work involves the use of netmap and netmap-fwd alongside the continues performance improvements done by the FreeBSD development branch.

# Network Concepts

#### **Network Performance Testing**

Tests the uplink and downlink speed of a network, which defines how quick and responsive a network is to I/O communication. Some of the main elements are:

Bandwidth: the volume of information per unit of time that a transmission medium (like an internet connection) can handle. It is usually expressed in bits per second.

Latency: the amount of time it takes a packet to travel from source to destination. Together, latency and bandwidth define the speed and capacity of a network.

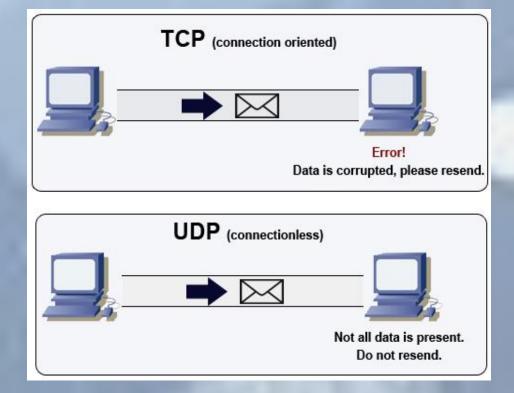


Figure 5. Diagram of TCP and UDP client/server interaction. From Tomasz Czyz

#### **Active Measurement Tools**

Tools like iperf3, nuttcp, netperf and others are used to measure TCP and UDP bandwidth performance. They create packets and transport them from client to server in order to saturate and measure the system performance. Some of the variables to play with include the packet size (64, 1500, 900 bytes), the number of streams, and the number of ports per test.

#### Tuning In order to achieve better performance in the system, variables such as the

### Router Interface

Hardware - Systems and NIC: The router is a Dell R530 with two Intel ® Xeon <sup>®</sup> E52695 CPUs, each one with 18 cores. Equipped with one T-580-CR 40 GbE 8 lane card.

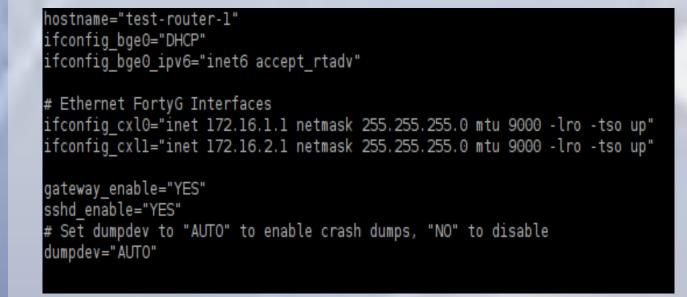


Figure 7. /etc/rc.conf gateway network configuration.

# Results

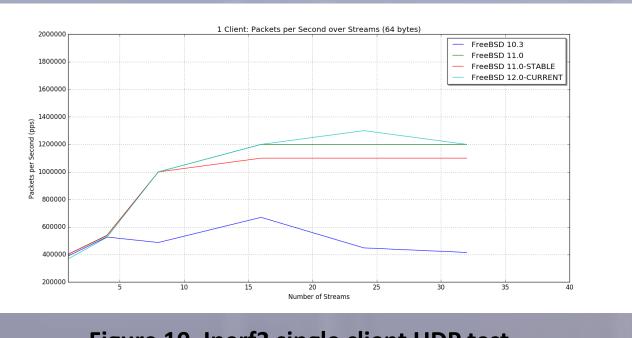


Figure 10. Iperf3 single client UDP test.

Graphs show how the newer versions of the operating system increase the amount of pps. One of the explanations to this event is the continuous improvements made by the FreeBSD developers branch towards performance and optimization.

FreeBSD has proven to be faster than Linux distributions. These systems are usually running fewer services, and packaged applications are configured by default with more performance tuning in mind.

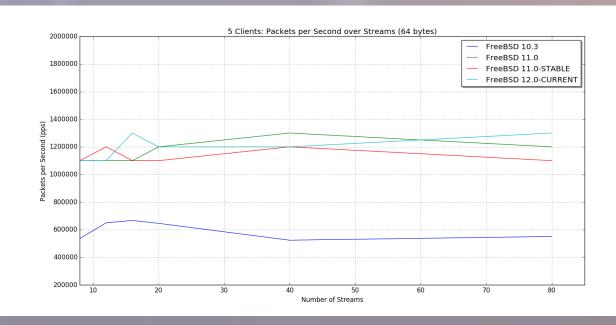


Figure 11. Iperf3 multi client UDP test.

# Results

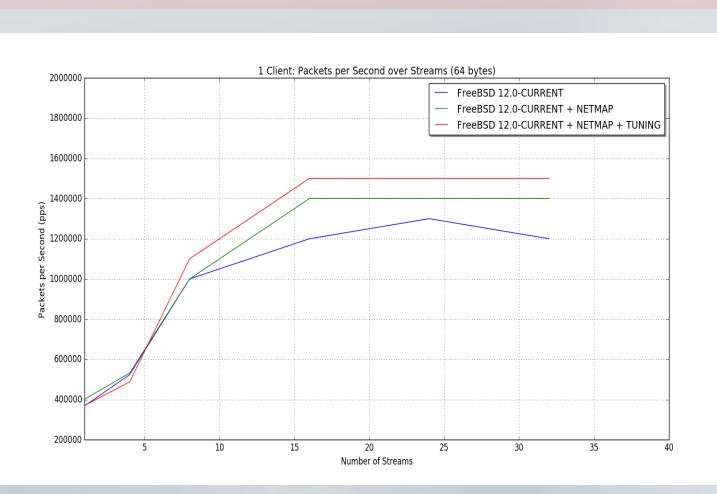


Figure 12. FreeBSD 12-CURRENT iperf3 single client UDP test.

It was proven that increasing the amount of streams does not necessarily raise pps. Actually, there is a stage in the system where the amount of pps stays constant. After adding some tuning variables scores increased over streams. However at this time interrupts were assigned to just one CPU for a 100%.

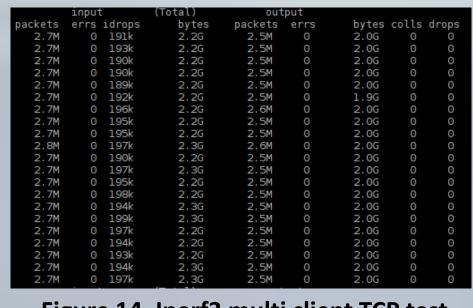


Figure 14. Iperf3 multi client TCP test.

FreeBSD 12.0-CURRENT branch proved to send more pps than the compared versions. Based on these results, a netmap interface was enable between clients and servers in the current version. Enabling this interface in the system increased significantly the

amount of pps sent as the number of streams increased.

Figure 13. FreeBSD 12-CURRENT iperf3 multi client UDP test.

Balancing the load to a specific amount of CPUs (approximately 8), raises the amount of pps as a factor of 1.7. By running this script interrupts have remained between 92% and 94%. Further tests need to be implemented in order to continue decreasing interrupts and increasing pps.

# Interesting Features

- Testing has shown that the amount of pps will rise as the newer versions of the operating system have been deployed.
- FreeBSD was able to send more pps as client than Centos 6.
- The choice of network card can have a significant impact on pps, tuning, and netmap support.
- Increasing the amount of streams and ports through time drops the packet amount. About 10 streams and 80 ports per client achieved the best scores.
- Netmap-fwd increases the amount of packets significantly.

# **Future Work**

- Enable the system with two 40G cards.
- Configure Lagg in the router switch ports.
- Use 40G clients and servers.
- Updates to the network capabilities in the FreeBSD-Current version will be closely monitored and applied as appropriate.
- The final result will be a reference architecture with representative hardware and software that will enable the NCCS to build, deploy, and efficiently maintain extremely cost-effective 100-Gbps firewalls.

# References

- FreeBSD forwarding Performance. Retrieved on August 22, 2016 from
- https://bsdrp.net/documentation/technical\_docs/performance
- ProjectRoutingProposal. Retrieved on September 18, 2016 from https://wiki.freebsd.org/ProjectsRoutingProposal/ConversionStatus#New\_routin g\_KPI\_conversion\_status
- Netmap. Retrieved on August 22,2016 from https://github.com/luigirizzo/netmap



Chelsio:

# Acknowledgements

- NASA Minority University Research and Education Scholarship.
- Thanks to George Rumney (mentor), John E. Jasen (mentor), Bennett Samowich, Hoot Thompson, and Benjamin Bledsoe from the NASA Center for Climate Simulations, for their continuous help during this work.
- Special thanks to the Chelsio support group for their assistance.

